Discrete Applied Mathematics 99 (2000) 261–277

DISCRETE
APPLIED
MATHEMATICS

# of their correlation coefficient

Eric Angel[a,*], Vassilis Zissimopoulos[b]

[a] *Université de Paris Sud, L.R.I, CNRS-URA 410, Centre d'Orsay, 91405 Orsay, France*
[b] *Université Paris Nord, LIPN, CNRS-URA 1507, 93430 Villetaneuse, France*

Received 9 October 1997; received in revised form 7 September 1998; accepted 9 March 1999

Abstract

Local search and its variants simulated annealing and tabu search are very popular meta-heuristics to approximatively solve NP-hard optimization problems. Several experimental studies in the literature have shown that in practice some problems (e.g. the Traveling Salesman Problem, Quadratic Assignment Problem) behave very well with these heuristics, whereas others do not (e.g. the Low Autocorrelation Binary String Problem). The autocorrelation function, introduced by Weinberger, measures the ruggedness of a landscape which is formed by a cost function and a neighborhood. We use a derived parameter, named the autocorrelation coefficient, as a tool to better understand these phenomena. In this paper we mainly study cost functions including penalty terms. Our results can be viewed as a first attempt to theoretically justify why it is often better in practice to enlarge the solution space and add penalty terms than to work solely on feasible solutions. Moreover, some new results as well as previously known results allow us to obtain a hierarchy of combinatorial optimization problems relatively to their ruggedness. Comparing this classification with experimental results reported in the literature yields a good agreement between ruggedness and difficulty for local search methods. In this way, we are also able to justify theoretically why a neighborhood is better than another for a given problem.

Keywords: Local search; Meta-heuristics; Autocorrelation coefficient; Classification of combinatorial optimization problems

1. Introduction

Consider the problem of minimizing a real-valued function C, over a finite and discrete search space $\mathscr{S}$. By definition, the *cost* of a solution $x \in \mathscr{S}$ is $C(x)$, and a solution which attains the minimum over $\mathscr{S}$ is called a *global minimum*.

* Corresponding author. Fax: +33 16915686.
E-mail addresses: angel@lri.fr (E. Angel), vz@lipn.univ-paris13.fr (V. Zissimopoulos)

0166-218X/00/\$ - see front matter
PII: S0166-218X(99)00138-9

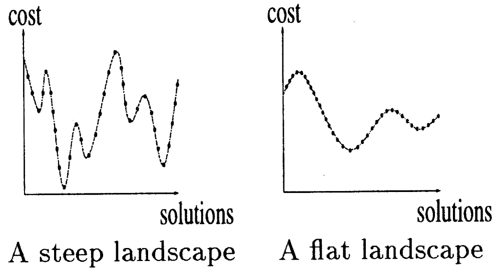

Fig. 1. Examples of landscapes.

To use local search one simply has to specify a *neighborhood structure* which associates for each solution $x \in \mathscr{S}$, a neighborhood $\mathscr{N}(x) \subset \mathscr{S}$. Then, a local search algorithm consists in iterating the following instruction, which has to take a polynomial time in order to be useful in practice: substitute the current solution x for a better one in its neighborhood $\mathscr{N}(x)$. The search will end to a *local optimum* (local minimum), that is, a solution for which none of its neighbors has a lower cost.

Therefore, when faced with a combinatorial optimization problem, one has to choose a neighborhood structure to use local search. This choice is of primary importance, but the cost function should also be considered. For example, one can choose to enlarge the solution space, allowing non-feasible solutions, and therefore add a penalty term in the cost function. The cost function to be optimized and the neighborhood form what is called a *landscape*. Of course, the quality of results obtained depends of these choices, leading to the notion of "well suited" and "bad suited" landscapes for local search algorithms.

The presence of numerous local minima in the landscape, represents the main obstacle for local search, and various generalizations have been developed to overcome this difficulty. Among the most popular are simulated annealing [1] and tabu search [12]. We call them generalized local search, but the number of local minima still remains the main difficulty they are faced with, and even if they are no more trapped in, they slow down the search.

One of the most important characteristics of a landscape is its ruggedness. There is a strong link between this concept and the hardness of an optimization problem relatively to a local search-based algorithm. Intuitively, it is clear that the number of local minima depends on the link between the cost of a solution and the cost of its neighbors. Therefore, a flat (respectively steep) landscape should be well (respectively bad) suited for a local search algorithm. Fig. 1 illustrates these notions.

The autocorrelation functions, introduced by Weinberger, measure the ruggedness of a landscape. The context of his work was to better understand evolutionary mechanisms in biology. Previously experimental results on the Quadratic Assignment Problem [4], have shown there was an excellent agreement between the value of the autocorrelation coefficient, which we derive from the autocorrelation functions, and the suitness of the landscape for local search algorithms.

In this paper we further study this link, specially on cost functions including penalty terms. Our results on some classical combinatorial optimization problems such as the Quadratic Assignment Problem [4], Max Cut, Node Cover, Weighted Independent Set Problem and Graph Bipartitioning Problem, as well as the previously known results on the Traveling Salesman Problem and Low Autocorrelation Binary String Problem, allow us to obtain a hierarchy of combinatorial optimization problems relatively to the ruggedness of their landscape according the neighborhood and cost function chosen. This classification is compared with previously experimental studies and it yields a good agreement between ruggedness of landscape and difficulty for local search-based algorithms.

2. The ruggedness of a landscape

How to define properly the ruggedness of a landscape? At first sight it seems natural to use the difference of cost between two neighboring solutions. Yet, let us consider Fig. 2 where two landscapes $(C,\mathcal{N})$ and $(aC,\mathcal{N})$, with $a\in\mathbb{R}$ a constant, are represented. These two landscapes should have the same ruggedness, and indeed it is equivalent from a difficulty point of view to optimize functions C and aC for local search based meta-heuristics.

In the following, we always have a symmetric neighborhood, i.e. $x\in\mathcal{N}(y) \Leftrightarrow y\in\mathcal{N}(x)$, for any two solutions $x,y\in\mathcal{S}$. Moreover, we suppose that each solution has the same number of neighboring solutions. We call this number the *size* of the neighborhood $\mathcal{N}$ and denote it by $|\mathcal{N}|$. Let the *distance* between any two distinct solutions x and y, denoted by $d(x,y)$, be the smallest integer $k\geqslant 1$ such that there exists a sequence of solutions $x_0,\dots,x_k$ with $x_0=x$, $\forall i\in\{0,\dots,k-1\}$, $x_{i+1}\in\mathcal{N}(x_i)$ and $x_k=y$. In the sequel, we always have $d(x,y)=d(y,x)$.

By definition, the landscape *autocorrelation function* [25] is

$$\rho(d)=1-\frac{\langle (C(x)-C(y))^2\rangle_{d(x,y)=d}}{\langle (C(x)-C(y))^2\rangle}$$

with $\langle (C(x)-C(y))^2\rangle$ the average value of $(C(x)-C(y))^2$ over all solutions pairs $\{x,y\}$, and $\langle (C(x)-C(y))^2\rangle_{d(x,y)=d}$ the average value of $(C(x)-C(y))^2$ over all solutions pairs $\{x,y\}$ which are at distance d. It is not difficult to see that

$$\rho(d)=1-\frac{\langle (C(x)-C(y))^2\rangle_{d(x,y)=d}}{2(\langle C^2\rangle-\langle C\rangle^2)}$$

with $\langle C\rangle$ (respectively $\langle C^2\rangle$) the average value of $C(x)$ (respectively $C^2(x)$) over $\mathcal{S}$. Notice that given a landscape, the autocorrelation function is non-random, i.e. its values are perfectly determined.

The quantity $\rho(d)$ shows the level of correlation between any two solutions which are at distance d from each other. The most important value to know is $\rho(1)$, because

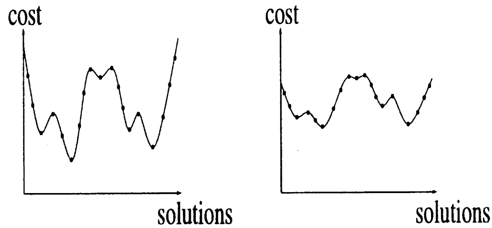

Fig. 2. Two landscapes $(C, \mathcal{N})$ and $(aC, \mathcal{N})$ with the same ruggedness.

the link between two adjacent solutions is of first importance for any local search-based meta-heuristic. A value close to 1 for $\rho(1)$ indicates that costs of any two neighboring solutions are on average very close. We shall consider that it means the landscape is flat. On the contrary, a value close to 0 indicates that the cost of any two neighboring solutions are almost independent, and we shall consider that it means a steep landscape. Notice that according to this definition the two landscapes pictured in Fig. 2 have the same ruggedness.

We define the autocorrelation coefficient ξ by $\xi = 1/(1-\rho(1))$. According to the previously comments, the larger ξ is, the more flat is the landscape, and the more suited is the landscape for any based local search heuristic.

Weinberger [25] also suggested to use random walks to investigate the correlation structure of a landscape. Consider the sequence of costs generated by a random walk (x_i), which at each step moves to a new solution chosen randomly among the neighbors of the current solution. One can define another autocorrelation function by putting

$$r(s) = 1 - \frac{\langle (C(x_i) - C(x_{i+s}))^2 \rangle}{2(\langle C^2 \rangle - \langle C \rangle^2)}$$

with s an integer. Notice the equality $\rho(1) = r(1)$.

It is possible to calculate autocorrelation functions when problems are randomly defined, that is to say when the cost C is a random variable. See [23] for a development of the theory, and [3] for a study concerning the Graph Bipartitioning Problem.

Closely related to a neighborhood structure is the *neighborhood graph* G whose vertex set is the set of solutions $\mathcal{S}$, and for which two solutions x and y are adjacent if and only if $y \in \mathcal{N}(x)$.

Given a landscape, let $G = (V, E)$ be its neighborhood graph. We shall suppose it is connected. We fix an arbitrary labeling for the vertices of G, and consider that the subsequent matrices and eigenvectors are indexed by V rather than integers. In the same way, $C(x)$ the cost of solution x, should be considered as the xth component of a vector C. Let D be the diagonal matrix of vertex degrees of G, and A the adjacency matrix of G. The matrix $-\Delta = D - A$ is called the *Laplacian* of G. It is symmetric, and hence there exists a complete orthonormal set of eigenvectors of $-\Delta$, denoted by $\{\phi_i\}_{i=0,\ldots,|V|-1}$. In fact, in the sequel we need a slighter condition: $\langle \phi_i, \phi_i \rangle = c$, with c a constant (not necessarily 1), and $\langle \phi_i, \phi_j \rangle = 0 \; \forall i \neq j$, with by definition $\langle \phi_i, \phi_j \rangle = \sum_{x \in V} \phi_i(x)\phi_j(x)$ the usual scalar product, and $\phi_i(x)$ the xth component of vector ϕ_i. Moreover, the eigenvector $\phi_0 = (1, \dots, 1)^{\mathrm{t}}$ has the eigenvalue $\lambda_0 = 0$ which has multiplicity 1. The Laplacian is non-negative definite, therefore $\lambda_i > 0$, for $i \in \{1, \dots, |V| - 1\}$.

By definition of vectors $\{\phi_i\}_{i=0,\dots,|V|-1}$ there always exists a decomposition $C(x) = \sum_{i=0}^{|V|-1} a_i \phi_i(x)$, with $a_i \in \mathbb{R}$. This decomposition is called a *Fourier expansion* of the landscape.

The main theorem concerning the autocorrelation function r due to Stadler [22], allows to calculate it, if one can write the cost function in terms of eigenvectors of the Laplacian of the neighborhood graph.

Theorem 1 (Stadler [22]). *Let $C = \sum_i a_i \phi_i$ be a Fourier expansion of a landscape $(C, \mathscr{N})$. Then its autocorrelation function r is equal to*

$$r(s) = \sum_{i \neq 0} \frac{a_i^2}{\sum_{j \neq 0} a_j^2} \left(1 - \frac{\lambda_i}{|\mathscr{N}|}\right)^s,$$

where λ_i is the eigenvalue associated with the eigenvector ϕ_i.

When, in a Fourier expansion of a landscape, all eigenvectors ϕ_i for $i > 0$ have the same eigenvalue λ, the landscape is called *elementary*. It was previously noticed by Grover in [13], that several combinatorial optimization problems verified the following equality $\forall x \in \mathscr{S}$, $\nabla^2(C)(x) + K(C(x) - \langle C \rangle) = 0$, with $K > 0$ a constant, $\langle C \rangle = (1/|\mathscr{S}|) \sum_{x \in \mathscr{S}} C(x)$ the average cost, and $\nabla^2(C)(x) = \sum_{i=1}^{|\mathscr{N}|} \delta_i / |\mathscr{N}|$, with $|\mathscr{N}|$ the neighborhood size, and δ_i the difference in cost between the current solution x and its ith neighboring solution. It was pointed out by Stadler that this was an equivalent definition of elementary landscapes. Indeed, one can observe that $\lambda = K\,|\mathscr{N}|$.

When a landscape is elementary, it follows from Theorem 1 that its autocorrelation function r is exponential, i.e. $r(s) = r(1)^s$ (and conversely). The *autocorrelation length* l is then defined by $r(s) = \rho(1)^s = \mathrm{e}^{-s/l}$, that is $l = -1/(\ln \rho(1))$. The larger is l, the closer to one is $\rho(1)$, and therefore the more suited for a local search is the landscape. Intuitively, the autocorrelation length l, indicates the minimum distance between any two solutions for them to have a non-correlated cost. Therefore, when one compares various elementary landscapes it is more rigorous to compare the ratios l/d, where d denotes the diameter of the neighborhood graph, than the values l. When $\rho(1) \to 1$ as the size of the instance $n \to \infty$, our autocorrelation coefficient ξ is asymptotically equal with the autocorrelation length l. Its interest is to be more general, as it is defined when the landscape is not elementary.

It is worthwhile to notice that, for all landscapes which have their cost function defined on boolean vectors, and whose neighborhood graph is the hypercube, by using Theorem 1 in conjunction with the following Proposition 1, we have a fast way to obtain the autocorrelation coefficient.

We consider that the vertices of the hypercube of dimension n are labeled $(\sigma_1, \sigma_2, \dots, \sigma_n)$ with $\forall i,\ \sigma_i \in \{-1, +1\}$. Let $\varepsilon_q(\sigma) = \sigma_{i_1} \cdot \sigma_{i_2} \cdots \sigma_{i_p}$, where $q = (q_1, \dots, q_n)$, and

$q_k = 1$ if $k \in \{i_1, \ldots, i_p\}$ and 0 otherwise. According to the convention stated above, $\varepsilon_q(\sigma)$ should be understood as the σth component of vector ε_q.

Proposition 1 (Stadler [22]). *The eigenvectors of the Laplacian of the boolean hypercube of dimension n are given by ε_q with q ranging over boolean multi-indices of length n. There is an exception when $q = (0, \ldots, 0)$ where for this case $\varepsilon^{\mathrm{t}}_{(0,\ldots,0)} = (1, \ldots, 1)$. Moreover, the eigenvalue corresponding to ε_q is always $2p$, with p the number of non-zero entries in the boolean multi-index q.*

3. Applications

3.1. Preliminaries

Let us consider some combinatorial optimization problems, and neighborhoods associated. In the sequel we have $x = (x_1, \ldots, x_n)$ with $x_i \in \{0, 1\}$, $\sigma = (\sigma_1, \ldots, \sigma_n)$ with $\sigma_i \in \{-1, 1\}$, π a permutation, and $C(x)$, $C(\sigma)$, or $C(\pi)$ the cost of a solution.

In the *flip* neighborhood we move from a solution x (or σ) to a neighboring one by changing the value of a single boolean x_i (or σ_i). In the *swap* neighborhood one moves from a solution x to a neighboring one by swapping a '0' bit with a '1' bit in x, i.e. we change the value of two booleans having distinct values. In the 2-*exchange* neighborhood, given a permutation $\pi = (\pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots, \pi(n))$, its neighbors are the $n(n-1)/2$ permutations of the form $(\pi(1), \ldots, \pi(j), \ldots, \pi(i), \ldots, \pi(n))$ for $1 \leqslant i < j \leqslant n$, obtained from π by performing a transposition.

Given n cities and a matrix $D = (d_{ij})$ representing distances between them, the Traveling Salesman Problem (TSP) asks to find a tour, of minimum length, passing through each city exactly once. We shall assume that D is symmetric with a null diagonal. For the TSP, we describe two possible neighborhoods. In the first one, called 2-opt, one moves from one circuit to a neighboring one by substituting 2 arcs in it with 2 arcs not in it. One defines in the same way, k-opt neighborhoods, with $k \geqslant 2$. In the 2-exchange neighborhood, one moves from one circuit to a neighboring one by exchanging the position of 2 cities in this circuit.

Given two $n \times n$ matrices $F = (f_{ij})$ and $D = (d_{ij})$, the Quadratic Assignment Problem (QAP) asks to find a permutation π which minimizes the sum $\sum_{ij} f_{ij} d_{\pi(i)\pi(j)}$. Notice that the QAP is a generalization of the TSP [5]. We shall assume that the matrices F and D are symmetric with a null diagonal. The 2-exchange neighborhood is the standard one for the QAP.

Given a complete graph with n (even) vertices, and a symmetric matrix $W = (w_{ij})$ of edge weights, the Graph Matching Problem (GMP) (in a complete weighted graph) asks to partition the vertices into $n/2$ pairs such that the sum of the edge weights corresponding to these pairs is minimum. Given a permutation π, a partition is obtained by assuming that the vertices $\pi(2i-1)$ and $\pi(2i)$ form a pair. Notice that there are several permutations which represent the same partition. We consider the 2-exchange neighborhood and the cost function is given by $C(\pi)=\sum_{i=1}^{n/2} w_{\pi(2i-1)\pi(2i)}$.

Given an edge-weighted graph, the Graph Bipartitioning Problem (GBP) asks to find a partition of its vertices into two equal-sized subsets, such that the total weight of edges connecting the two subsets (the *cutweight*) is minimum. We note $w_{ij}\in\mathbb{R}$ the weight of the edge between vertices i and j ($w_{ii}=0$). Without loss of generality, we can suppose that the graph is complete by putting $w_{ij}=0$ when the edge does not exist. We put $x=(x_1,\dots,x_n)$, with $x_i=1$ (respectively $x_i=0$) meaning that vertex i belongs to the first (respectively second) subset. In [16] (see also the companion paper [17]) Johnson et al. report an extensive empirical study for simulated annealing applied to the unweighted version of this problem, i.e. w_{ij} can take only values 0 or 1. They have considered the swap neighborhood and the flip neighborhood. In the first one, only equal sized partitions of the vertex set are considered, and the cost function is $C(x)=\sum_{i,j} w_{ij}x_i(1-x_j)$, for all x that satisfy $\sum_i x_i=n/2$.

In the second one, which we call the α-flip neighborhood, any partition of the vertex set is a solution, and to penalize non-equal sized partitions a penalty term, which is a function of a coefficient α called the *imbalance factor*, is added to the cost function. The cost function is therefore $C(x)=\sum_{i,j} w_{ij}x_i(1-x_j)+\alpha(\sum_i 2x_i-n)^2$. When the simulated annealing ends the following greedy heuristic is repeated until one obtains an equal sized partition: find a vertex in the larger set that can be moved to the other set with the least increase in the cutweight, and move it.

Given an edge-weighted graph, the Max Cut (MC) problem asks to find a subset of the vertices that maximizes the weights of edges having one extremity selected and the other not selected. A solution is represented by a boolean vector x, with the meaning that $x_i=1$ if vertex i is selected, and 0 otherwise. The flip neighborhood is used and the cost function is $C(x)=\sum_{i,j} w_{ij}x_i(1-x_j)$. For the MC problem, a Boltzmann machine is developed in [1], with a consensus function equal to this cost function.

Given a graph G with weights associated to vertices, the Node Cover Problem (NC) asks to find a minimum vertex cover, that is a minimum weighted subset of vertices such that each edge of G has at least one extremity in it. We use the flip neighborhood (more precisely we denote it by α-flip), and the cost function which has to be minimized is $C(x)=\sum_i x_iw_i+\alpha\sum_{i,j}(1-x_i)(1-x_j)d_{ij}$. In other words, one seeks a subset of vertices which minimizes the sum of the weight of vertices in that subset while penalizing uncovered edges.

Given a graph with weights associated to vertices, the Weighted Independent Set (WIS) problem asks to select a subset of vertices, such that no two selected vertices are adjacent, and such that the sum of their weights is maximized. The flip neighborhood is used. We note $D=(d_{ij})$ the adjacency matrix of the graph, and the cost function which has to be maximized is $C(x)=\sum_i x_iw_i-(\alpha/2)\sum_{i,j} x_ix_jd_{ij}$. Notice that there is a penalty term function of a coefficient α, used to penalize solutions with a lot of induced edges.

In [1] a Boltzmann machine is developed for this problem, and the consensus function C is a special case of our cost function with $\forall i$, $w_i=\beta>0$ and $\alpha>\beta$. With these conditions, the consensus function is *feasible* and *order-preserving*, and it means that all local optima of C correspond to feasible solutions, and that if a solution has an independent set larger than another solution, then its consensus is better than the consensus of the other solution. In case the cost function is not feasible, a greedy heuristic is used to obtain a feasible solution.

Given n reals, $w_1, \dots, w_n$, the Weight Partition Problem (WPP) asks to find a binary vector σ, over the alphabet $\{-1, +1\}$, which minimizes the sum $C(\sigma) = (\sum_i \sigma_i w_i)^2$. The flip neighborhood is used. This problem is equivalent with the $P2||C_{\max}$ problem which consists in scheduling n jobs $i=1, \dots, n$, with processing time w_i, on two identical parallel machines I_1 and I_2, such that the makespan is minimized.

We finally introduce two more problems on which the flip neighborhood is used.

Given a set of clauses (disjunction of three literals), the Not-All-Equal-Satisfiability problem (NAES) asks to find an assignment maximizing the number of satisfied clauses. By definition, a clause is said to be satisfied if all its three literals do not have the same value.

The Low Autocorrelation Binary String problem (LABS) asks to find binary sequences σ over the alphabet $\{-1, +1\}$, which have minimum off-peak autocorrelation coefficients $R(k)$, where by definition $R(k) = \sum_{i=1}^{n-k} \sigma_i \sigma_{i+k}$. The cost function, which has to be minimized, is $C(\sigma) = \sum_{k=1}^{n-1} R(k)^2$. This problem has technical applications such as synchronization in digital communication systems and modulation of radar pulses [6].

The Graph c-Coloring problem (GC(c)) asks to color a graph with c colors such that the number of edges whose vertices have the same color is minimized. In the flip neighborhood one moves from a solution to a neighboring one by changing the color of a vertex.

All these problems are NP-hard, except GMP which is polynomial [10], and LABS for which we are not aware of a proof of its NP-hardness.

3.2. Results

The TSP with 2-opt and 2-exchange neighborhoods, GMP with the 2-exchange neighborhoods, GBP with the swap neighborhood, GC(c) with the c-flip neighborhood, NAES, WPP, LABS with the flip neighborhood had been previously considered in [13,8,22,24]. In [4] we have proved that the autocorrelation coefficient of any instance of the QAP, with the 2-exchange neighborhood, verifies $\xi \geqslant n/4$. Here, we undertake a systematic presentation of these known results and we add results on GBP with the α-flip neighborhood, MC with the flip neighborhood, and WIS with the α-flip neighborhood, in order to clarify the suitness of local search-based methods for solving these problems, as well as to better choose the value of the penalty factor α which is in practice chosen purely experimentally. For each of these problems we show how to use the above theorem due to Stadler to obtain the autocorrelation coefficient ξ. In the appendix we present analytical formula for $\langle C \rangle$, $\langle C^2 \rangle$, and $\langle (C(x) - C(y))^2 \rangle_{d(x,y)=1}$, allowing us to obtain the autocorrelation coefficient as well.

We consider first the Graph Bipartitioning Problem with the α-flip neighborhood. We note w_{ij} the edge weight between vertices i and j. Let W_k denote the sum $\sum_{i,j} w_{ij}^k$, and w_k the sum $\sum_i w_{ki}$. We shall note $|x|$ for $\sum_i x_i$.

Proposition 2. *For any α, the autocorrelation coefficient of any instance of GBP with the α-flip neighborhood, is $n/4$.*

Proof. The cost function is $C(x) = \sum_{i,j} w_{ij}x_i(1-x_j) + \alpha(2|x|-n)^2$.
Then, by putting $\sigma_i = 2x_i - 1$, we obtain

$$C(\sigma) = \sum_{i,j} \frac{w_{ij}}{4} + \sum_{i \neq j} \left(\alpha - \frac{w_{ij}}{4}\right) \sigma_i \sigma_j$$

and by using Theorem 1 and Proposition 1 we have $r(s) = (1 - 4/n)^s$ and $\xi = 1/(1 - r(1)) = n/4$. Notice that this landscape is elementary. □

Proposition 3. *The autocorrelation coefficient of any instance of MC with the flip neighborhood, is $n/4$.*

Proof. The cost function corresponds to the special case of graph bipartitioning problem with the α-flip neighborhood when $\alpha = 0$. The fact that the problem is a maximization one, whereas the graph bipartitioning problem is a minimization one, does not change the value of the autocorrelation coefficient. □

Proposition 4 (Stadler [22]). *The autocorrelation coefficient of any instance of the Weighted Partition Problem, with the flip neighborhood, is $n/4$.*

Proof. The cost function is $C(\sigma) = (\sum_i \sigma_i w_i)^2 = \sum_i w_i^2 + \sum_{i \neq j} w_i w_j \sigma_i \sigma_j$, and by using Theorem 1 and Proposition 1 we obtain $\xi = n/4$. □

We now consider the Weighted Independent Set problem. Let $D = (d_{ij})$ be the adjacency matrix of the graph, and let d_i (respectively w_i) be the degree (respectively weight) of vertex i.

Proposition 5. *The autocorrelation coefficient of an instance of WIS, with the α-flip neighborhood, is given by $\xi = n/2$ if $\alpha = 0$, and*

$$\xi = \frac{n}{2}\left(1 - \frac{1}{2(1 + \sum_i d_i^2 / \sum_i d_i) - (8/\alpha)\sum_i d_i w_i / \sum_i d_i + (8/\alpha^2)\sum_i w_i^2 / \sum_i d_i}\right)$$

if $\alpha > 0$.

Proof. The cost function is $C(x)=\sum_i x_i w_i-(\alpha/2)\sum_{i,j} x_i x_j d_{ij}$. By putting $\sigma_i=2x_i-1$, we obtain

$$\begin{aligned} C(\sigma) &= \sum_i \frac{\sigma_i+1}{2} w_i - \frac{\alpha}{2}\sum_{i,j}\frac{\sigma_i+1}{2}\frac{\sigma_j+1}{2} d_{ij} \\ &= \frac{1}{2}\sum_i w_i + \frac{1}{2}\sum_i w_i\sigma_i - \frac{\alpha}{2}\left(\frac{1}{4}\sum_{i,j} d_{ij}\sigma_i\sigma_j + \frac{1}{2}\sum_i d_i\sigma_i + \frac{1}{4}\sum_{i,j} d_{ij}\right) \\ &= \frac{1}{2}\sum_i w_i - \frac{\alpha}{8}\sum_{i,j} d_{ij} + \sum_i\left(\frac{1}{2}w_i - \frac{\alpha}{4}d_i\right)\sigma_i - \frac{\alpha}{4}\sum_{i<j} d_{ij}\sigma_i\sigma_j. \end{aligned}$$

According to Proposition 1 this is a Fourier decomposition of the landscape: $\{\sigma_i\}$ are eigenvectors of the Laplacian of the hypercube with eigenvalue 2, and $\{\sigma_i\sigma_j\}_{i<j}$ are eigenvectors with eigenvalue 4.

By applying Theorem 1 one obtains

$$\begin{aligned} r(s) &= \frac{\sum_i w_i^2/4 + (\alpha^2/16)d_i^2 - (\alpha/4)w_i d_i}{\sum_i w_i^2/4 + (\alpha^2/16)d_i^2 - (\alpha/4)w_i d_i + \sum_{i<j}(\alpha^2/16)d_{ij}^2}\left(1-\frac{2}{n}\right)^s \\ &\quad + \frac{\sum_{i<j}(\alpha^2/16)d_{ij}^2}{\sum_i w_i^2/4 + (\alpha^2/16)d_i^2 - (\alpha/4)w_i d_i + \sum_{i<j}(\alpha^2/16)d_{ij}^2}\left(1-\frac{4}{n}\right)^s. \end{aligned}$$

Using the fact that $d_{ij}^2=d_{ij}$ as $d_{ij}\in\{0,1\}$, one obtains after some simplifications the theorem. □

Proposition 6. *The autocorrelation coefficient of any instance of the WIS problem, with the α-flip neighborhood for any $\alpha\geqslant 0$, verifies $\xi\geqslant n/4$.*

Proof. We note $\xi(\alpha)$ the autocorrelation coefficient of the α-FLIP landscape. By using Proposition 5 we obtain

$$\xi(\alpha)\geqslant\frac{n}{4} \;\Leftrightarrow\; \underbrace{2\frac{\sum_i d_i^2}{\sum_i d_i} - \frac{8}{\alpha}\frac{\sum_i d_i w_i}{\sum_i d_i} + \frac{8}{\alpha^2}\frac{\sum_i w_i^2}{\sum_i d_i}}_{f(\alpha)} \geqslant 0.$$

Taking the derivative of f, one obtains that its minimum is attained when $\alpha=\alpha_{\min}=2\sum_i w_i^2/\sum_i d_i w_i$. Therefore, we have

$$\begin{aligned} \forall\alpha\geqslant 0,\ \xi(\alpha)\geqslant\frac{n}{4} &\Leftrightarrow f(\alpha_{\min})\geqslant 0 \\ &\Leftrightarrow \left(\sum_i d_i^2\right)\left(\sum_i w_i^2\right) - \left(\sum_i d_i w_i\right)^2 \geqslant 0, \end{aligned}$$

after some calculus. This last inequality is always true due to Hölder's inequality. Recall that given positive reals $a_1,\ldots,a_n,b_1,\ldots,b_n$, p,q such that $1/p+1/q=1$, Hölder's inequality states that $\sum_i a_i b_i \leqslant (\sum_i a_i^p)^{1/p} (\sum_i b_i^q)^{1/q}$. In the proof we have used the special case $p = q = 2$. □

Proposition 7. *The autocorrelation coefficient of any instance of the NC problem, with the* α*-flip neighborhood for any* $\alpha \geqslant 0$, *verifies* $\xi \geqslant n/4$.

Proof. We have $\xi_{NC}(\alpha) = \alpha \sum_{i,j} d_{ij} + \sum_i (w_i - 2\alpha d_i) x_i + \alpha \sum_{i,j} x_i x_j d_{ij}$. Notice that this cost function can be obtained from the cost function of the WIS problem by adding a constant term (independent of the solution x) $\alpha \sum_{i,j} d_{ij}$ and by substituting α by -2α and w_i by $w_i - 2\alpha d_i$ in the cost function C_{WIS}. Since adding a constant term to the cost function does not change the autocorrelation coefficient, one can readily obtain by making the above proper substitutions the expression of the autocorrelation coefficient for the NC problem from Proposition 5. Now applying Proposition 6, which is also true for negative weights w_i and negative α, leads to the result. □

When the cost functions are defined on permutations, and the landscape is not elementary, like the Quadratic Assignment Problem, then it seems much more difficult (if not impossible) to apply Theorem 1 (see [4]).

4. Classification

The results are summarized in Table 1.

The high value of ξ for TSP (2-opt) suggests that local search-based heuristics are well adapted for this problem. In fact, the experimental study presented by Johnson in [15] confirms this: “An inescapable general conclusion, in light of the results presented here for approximation algorithms [...], is that the TSP is in practice much less formidable than its reputation would suggest”.

The same conclusion is obtained for the QAP and two experimental studies confirm this fact. One can read in [20] “[...] reflects a QAP property which was not yet put into evidence in the literature: any approach based on local search is bound to be very effective as a heuristic for QAP”, and in [9] “Simulated annealing is an extremely efficient heuristic for the QAP”.

On the contrary the low value for the LABS problem suggests that any local search-based heuristic will lead to poor results. The experimental study in [6] confirms this fact: “The reason why the promising statistical cooling method does not yield significant improvements can be the fact that it is often the case that a sequence with a high value of the merit factor has only neighbors with a low value of the merit factor. For such combinatorial optimization problems statistical cooling does not seem to be very well-suited for finding the global optimum”.

For the TSP, one can conclude that the 2-opt neighborhood ($\xi = n/2$) is better than the 2-exchange neighborhood ($\xi = n/4$), an experimentally well-known fact [7].

Table 1
Classification of problems according to their autocorrelation coefficient ξ

ξ	prb	$\mathscr{N}$	Elementary	$\lvert\mathscr{N}\rvert$	$\lvert\mathscr{S}\rvert$	Diameter	Reference
$n/2$	TSP	2-opt	Yes	$\frac{n(n-3)}{2}$	$\frac{(n-1)!}{2}$	$n-2$	[8]
$\geqslant n/4$	QAP	2-exchange	No	$\frac{n(n-1)}{2}$	$n!$	$n-1$	[4]
$n/4$	TSP	2-exchange	Yes	$\frac{n(n-1)}{2}$	$\frac{(n-1)!}{2}$	$n-2$	[8]
$(n+1)/6$	TSP	2,3-exchange	Yes	n^3-n	$\frac{(n-1)!}{2}$	$\leqslant n-2$	[8]
$\frac{c-1}{c}n/2$	GC(c)	c-FLIP	Yes	$n(c-1)$	c^n	n	[13]
$\geqslant n/4$	WIS	α-FLIP	No	n	2^n	n	This paper
$\geqslant n/4$	NC	α-FLIP	No	n	2^n	n	This paper
$n/4$	GBP	α-FLIP	Yes	n	$\binom{n}{n/2}$	n	This paper
	MC	FLIP	Yes	n	2^n	n	This paper
	NAES	FLIP	Yes	n	2^n	n	[13]
	WPP	FLIP	Yes	n	2^n	n	[22] and [13]
	GMP	2-exchange	Yes	$\frac{n(n-1)}{2}$	$\frac{n!}{(n/2)!2^{n/2}}$	$(n/2)-1$	[24]
$\simeq n/8$	GBP	SWAP	Yes	$\frac{n^2}{4}$	$\binom{n}{n/2}$	$n/2$	[22]
	LABS	FLIP	No	n	2^n	n	[22]

For the GBP, a value of $n/4$ for autocorrelation coefficient for the α-flip neighborhood and $n/8$ for the swap neighborhood, is a justification that it is better to allow non feasible solutions (non-balanced partitions of the vertex set) in conjunction with a penalty term in the cost function, than to just work with feasible solutions (balanced partitions), as it was noticed by Johnson et al. in their extensive experimental study [16].

Unfortunately, for the GBP with the α-flip neighborhood, the autocorrelation coefficient does not allow one to obtain information about the penalty factor α.

However, the situation is different, for the WIS problem. If we consider the autocorrelation coefficient as a function of α, $\xi(\alpha)$, then a straight calculation shows that its minimum is attained when $\alpha = 2\sum_i w_i^2 / \sum_i d_i w_i$ and therefore, a region of values around it should be avoided in practice.

There are several problems which have an autocorrelation coefficient equal to $n/4$. It is difficult to make a conclusion for these intermediate problems. Johnson et al. have studied the WPP problem with simulated annealing [17], but they have considered $|\sum_i \sigma_i w_i|$ for the cost function whereas we have used $(\sum_i \sigma_i w_i)^2$. Nevertheless, we cite an abstract of their experimental study as it confirms the importance of the ruggedness concept for the performance of local search algorithms: “The challenge of this problem is that the natural “neighborhood structures” for it, those in which neighboring solutions differ with regard to the location of only one or two elements, have exceedingly ‘mountainous’ terrain, in which neighboring solutions differ widely in quality. Thus, traditional local optimization algorithms are not competitive with other techniques for this problem, [...]”.

A promising direction of research would be to establish a link between the ruggedness and the NP-hardness of problems. The GMP is instructive for that purpose. It is a problem which has a less rugged elementary landscape compared to its graph diameter neighborhood, and it is the only one for which a polynomial algorithm is known. Is there a link? Weinberger explores related subjects in [26].

5. Future work

Two questions naturally arise: how could the ruggedness of a landscape be defined in another way, and what are its others important features?

Finding an alternative definition would be of interest, because, nevertheless, its advantages the autocorrelation coefficient does not allow to make a distinction between for example all instances of the TSP problem with the 2-opt neighborhood.

For the second question two previously introduced parameters seem relevant to us. The *depth* primarily introduced (not explicitly) by Hajek [14] and further studied by Kern [18] (see also [19]), and the *width* introduced by Ryan [21]. In the case of a minimization problem, the depth of a strict local (not global) minimum x is the least number d such that it is possible to reach a better local minimum y starting from x and moving through intermediate solutions z which verifies $C(z) < C(x) + d$. The width is defined in a similar way by considering the number of moves which deteriorate the solution. The depth (respectively width) of a landscape is the maximum depth (respectively width) of its local minima. Hajek's Theorem establishes for simulated annealing a link between the depth and a cooling schedule ensuring an asymptotic convergence towards a global minimum. Obtaining similar results for the ruggedness of landscapes would be of course of primary interest. Intuitively, a rugged landscape should need a slower decreasing temperature schema than a flat one.

The ruggedness of a landscape could also be useful when choosing the function to optimize for a given problem (consider for example the WPP problem).

Finally, defining a ruggedness measure when cost functions are continuous is a natural extension of this work. Indeed, meta-heuristics are sometimes used to optimize continuous problems [11].

Acknowledgements

The authors are grateful to the referees for their interesting suggestions and remarks contributing to improve the presentation of the paper.

Appendix

As we have said before, the theorem of Stadler does not apply for some problems. For these problems, like the Quadratic Assignment Problem, we have developed in [2,4] an alternative method for calculating the autocorrelation coefficient ξ which is more general. It consists of calculating in a direct way the quantities involved in the definition of the autocorrelation coefficient: $\langle C\rangle$, $\langle C^2\rangle$ and $\langle (C(x)-C(x'))^2\rangle$. The expressions for some problems studied in this paper are given in the following propositions. The complete proofs can be found in [2]. We give below the proof of Proposition 10 as an illustration of this method.

Proposition 8 (Stadler [2]). *For the Graph Bipartitioning Problem, we note w_{ij} for the weight of an edge between vertices i and j. Let W_k denote the sum $\sum_{i,j} w_{ij}^k$. Then, the average value of C, C^2 and the average squared cost difference between two neighboring solutions, for the α-flip neighborhood, are given by*

$$\langle C\rangle = \frac{W_1}{4} + \alpha n, \quad \langle C^2\rangle = \frac{1}{16}(W_1^2 + 2W_2) + \left(\frac{n}{2} - 1\right)\alpha W_1 + \alpha^2(3n^2 - 2n)$$

and

$$\langle (C(x) - C(x'))^2\rangle = \frac{W_2}{n} - \frac{8\alpha}{n} W_1 + 16(n-1)\alpha^2.$$

Proposition 9 (Stadler [2]). *For the Weighted Partition Problem, let W_k the sum $\sum_{i=1}^n w_i^k$. Then we have with the α-flip neighborhood*: $\langle C\rangle = W_2$, $\langle C^2\rangle = 3W_2^2 - 2W_4$, and $\langle (C(x)-C(x'))^2\rangle = (16/n)(W_2^2 - W_4)$.

Proposition 10 (Angel and Zissimopoulos [2]). *The average cost of C, C^2 and the average squared cost difference between two neighboring solutions, for the WIS problem with the α-flip neighborhood, are given by*

$$\langle C\rangle = \frac{W_1}{2} - \alpha\frac{D_1}{8},$$

$$\langle C^2\rangle = \frac{W_1^2 + W_2}{4} + \frac{\alpha^2}{4}\left(\frac{D_1^2 + 2D_1}{16} + \frac{1}{4}\sum_i d_i^2\right) - \alpha\left(\frac{W_1 D_1}{8} + \frac{1}{4}\sum_{i\neq j} w_i d_{ij}\right)$$

and

$$\langle (C(x) - C(x'))^2\rangle = \frac{W_2}{n} + \frac{\alpha^2}{4n}\left(\sum_i d_i + \sum_i d_i^2\right) - \frac{\alpha}{n}\sum_i d_i w_i.$$

Proposition 11 (Angel and Zissimopoulos [4]). *For the Quadratic Assignment Problem, let $F_k = \sum_{i,j} f_{ij}^k$, f_i for the sum $\sum_j f_{ij}$, D_k for the sum $\sum_{i,j} d_{ij}^k$, and d_i for the sum $\sum_j d_{ij}$. Then, the variance* $\mathrm{Var}(C) = \langle C^2\rangle - \langle C\rangle^2$ *of the cost function and the average squared cost difference between two neighboring solutions, for the* 2*-exchange neighborhood, are given by*

$$\mathrm{Var}(C) = \Bigg\{ F_2 D_2 n^4 - 2\left(F_2\left(2D_2 + \sum_i d_i^2\right) + \sum_i f_i^2\left(D_2 - \sum_i d_i^2\right)\right) n^3$$

$$+ \left(F_1^2\left(D_2 - 2\sum_i d_i^2\right) + F_2\left(D_1^2 + 5D_2 + 4\sum_i d_i^2\right)\right.$$

$$
-2\sum_i f_i^2(D_1^2-2D_2)\Bigg)n^2+\left(F_1^2\left(2D_1^2-D_2+2\sum_i d_i^2\right)\right.
$$

$$
-F_2\left(D_1^2+2\left(D_2+\sum_i d_i^2\right)\right)+2\sum_i f_i^2\left(D_1^2-D_2-\sum_i d_i^2\right)\Bigg)n
$$

$$
-3F_1^2D_1^2\Big\}\Big/(2n^2(n-1)^2(n-2)(n-3))
$$

and

$$
\begin{aligned}
\langle (C(x)-C(x'))^2\rangle = 4\Bigg(&F_2D_2n^3-\Bigg(2F_2\left(2D_2+\sum_i d_i^2\right)\\
&+\sum_i f_i^2\left(2D_2-\sum_i d_i^2\right)\Bigg)n^2+\Bigg(F_1^2\left(D_2-\sum_i d_i^2\right)\\
&+F_2\left(D_1^2+5D_2+4\sum_i d_i^2\right)\\
&-\sum_i f_i^2\left(D_1^2-4D_2-3\sum_i d_i^2\right)\Bigg)n\\
&+F_1^2\left(D_1^2-D_2-\sum_i d_i^2\right)-\left(D_1^2+2\left(D_2+\sum_i d_i^2\right)\right)\\
&\times\left(F_2+\sum_i f_i^2\right)\Bigg)\Big/(n^2(n-1)^2(n-2)(n-3)).
\end{aligned}
$$

Proposition 12 (Angel and Zissimopoulos [4]). *The autocorrelation coefficient of any instance of the QAP, with the 2-exchange neighborhood, verifies* $\xi \geqslant n/4$.

Proof of Proposition 10. Let $D=(d_{ij})$ be the adjacency matrix of the graph, and let d_i (respectively w_i) be the degree (respectively weight) of vertex i. We note W_k the sum $\sum_i w_i^k$, and D_k the sum $\sum_{ij} d_{ij}^k$. The cost function is $C(x)=\sum_i x_iw_i-(\alpha/2)\sum_{i,j} x_ix_jd_{ij}$.

Lemma 1. *The average cost is given by* $\langle C\rangle = W_1/2-\alpha D_1/8$.

Proof. We have

$$
\langle C\rangle = \frac{1}{2^n}\left(\sum_x\sum_i x_iw_i-\frac{\alpha}{2}\sum_x\sum_{i,j} x_ix_jd_{ij}\right)
$$

$$
=\frac{1}{2^n}\sum_i w_i\sum_x x_i-\frac{1}{2^n}\frac{\alpha}{2}\sum_{i\neq j} d_{ij}\sum_x x_ix_j
$$

$$= \frac{1}{2^n} 2^{n-1} W_1 - \frac{1}{2^n} \frac{\alpha}{2} 2^{n-2} D_1$$

$$= \frac{W_1}{2} - \alpha \frac{D_1}{8}. \qquad \square$$

The average squared cost $\langle C^2 \rangle$ can be computed also directly by similar calculations. We note $e_r = (0, \dots, 0, 1, 0, \dots, 0)$ the vector with the 1 at the rth position.

Lemma 2. *We have for the average squared cost difference between two neighboring solutions*

$$\langle (C(x) - C(x'))^2 \rangle = \frac{1}{n2^{n-1}} \sum_r \sum_{x:x_r=1} (C(x) - C(x - e_r))^2.$$

Proof. Given two neighboring solutions x and x', we can always order them such that there exists $1 \leqslant r \leqslant n$ such that $x' = x - e_r$, i.e. $\{x, x'\}$ forms an edge in the hypercube of dimension n which has $n2^{n-1}$ edges. $\square$

Lemma 3. *The average squared cost difference between two neighboring solutions is given by*

$$\langle (C(x) - C(x'))^2 \rangle = \frac{W_2}{n} + \frac{\alpha^2}{4n} \left(\sum_i d_i + \sum_i d_i^2 \right) - \frac{\alpha}{n} \sum_i d_i w_i.$$

Proof. We have

$$\sum_{x:x_r=1} (C(x) - C(x - e_r))^2 = \sum_{x:x_r=1} \left(w_r - \alpha \sum_i x_i x_r d_{ir} \right)^2$$

$$= \sum_{x:x_r=1} w_r^2 - 2\alpha \sum_i w_r d_{ir} \sum_{x:x_r=1} x_i + \alpha^2 \sum_{x:x_r=1} \left(\sum_i x_i d_{ir} \right)^2$$

$$= 2^{n-1} w_r^2 - 2\alpha 2^{n-2} w_r d_r + \alpha^2 \sum_{i,j} d_{ir} d_{jr} \sum_{x:x_r=1} x_i x_j$$

$$= 2^{n-1} w_r (w_r - \alpha d_r) + \alpha^2 \left(\sum_{i \neq j} 2^{n-3} d_{ir} d_{jr} + \sum_i 2^{n-2} d_{ir} \right)$$

$$= 2^{n-1} w_r (w_r - \alpha d_r) + \alpha^2 \left(2^{n-3} \sum_{i,j} d_{ir} d_{jr} + \sum_i 2^{n-3} d_{ir} \right)$$

$$= 2^{n-1} w_r (w_r - \alpha d_r) + \alpha^2 2^{n-3} (d_r^2 + d_r)$$

and the result follows by using Lemma 2. $\square$

References

[1] E. Aarts, J. Korst, Simulated Annealing and Boltzman Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing, Wiley, New York, 1989.
[2] E. Angel, V. Zissimopoulos, Towards a classification of combinatorial optimization problems relatively to their difficulty for generalized local search algorithms, Technical Report 1112, LRI, Université de Paris-Sud, 1997.
[3] E. Angel, V. Zissimopoulos, Autocorrelation coefficient for the graph bipartitioning problem, Theoret. Comput. Sci. 191 (1998) 229–243.
[4] E. Angel, V. Zissimopoulos, On the landscape ruggedness of the quadratic assignment problem, Technical Report 98-02, Université de Paris-Nord, LIPN, Institut Galilée, 1998.
[5] E. Angel, V. Zissimopoulos, On the quality of local search for the quadratic assignment problem, Discrete Appl. Math. 82 (1998) 15–25.
[6] G.F.M. Beenker, T.A.C.M. Claasen, P.W.C. Hermens, Binary sequences with a maximally flat amplitude spectrum, Philips J. Res. 40 (1985) 289–304.
[7] V. Černy, A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, J. Optim. Theory Appl. 45 (1985) 41–51.
[8] B. Codenotti, L. Margara, Local properties of some NP-complete problems, Technical Report TR-92-021, International computer science institute, Berkeley, CA 94704, April 1992.
[9] D.T. Connolly, An improved annealing scheme for the qap, European J. Oper. Res. 46 (1990) 93–100.
[10] J. Edmonds, Paths, trees, and flowers, Can. J. Math. 17 (1965) 449–467.
[11] M.A. Fleischer, Generalized cybernetic optimization: solving continuous variable problems, in second International Conference on Metaheuristics, Sophia-Antipolis, France, July 21–24, 1997.
[12] F.W. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Dordrecht, 1997.
[13] Lov K. Grover, Local search and the local structure of NP-complete problems, Oper. Res. Lett. 12 (1992) 235–243.
[14] B. Hajek, Cooling schedules for optimal annealing, Math. Oper. Res. 13 (1987) 311–329.
[15] D.S. Johnson, Local optimization and the traveling salesman problem, ICALP 90 Automata, Languages and Programming, 1990, pp. 446–461.
[16] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Optimization by simulated annealing: an experimental evaluation; Part I, Graph partitioning, Oper. Res. 37 (6) (1989) 865–892.
[17] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Optimization by simulated annealing: an experimental evaluation; Part II, Graph coloring and number partitioning, Oper. Res. 39 (1991) 378–406.
[18] W. Kern, On the depth of combinatorial optimization problems, Discrete Appl. Math. 43 (1993) 115–129.
[19] M. Lundy, A. Mees, Convergence of an annealing algorithm, Math. Program. 34 (1986) 111–124.
[20] V. Maniezzo, M. Dorigo and A. Colorni, Algodesk: an experimental comparison of eight evolutionary heuristics for the quadratic assignment problem, European J. Oper. Res. 81 (1995) 188–204.
[21] J. Ryan, The depth and width of local minima in discrete solution spaces, Discrete Appl. Math. 56 (1995) 75–82.
[22] P.F. Stadler, Landscapes and their correlation functions, Technical Report 95-07-067, Santa Fe institute, Santa Fe, NM, 1995.
[23] P.F. Stadler, Towards a theory of landscapes, Technical Report 95-03-030, Santa Fe Institute, Santa Fe, NM, 1995.
[24] P.F. Stadler, R. Happel, Canonical approximation of landscapes, Technical Report 94-09-051, Santa Fe institute, Santa Fe, NM, 1994.
[25] E.D. Weinberger, Correlated and uncorrelated fitness landscapes and how to tell the difference, Biol. Cybernet. 63 (1990) 325–336.
[26] E.D. Weinberger, NP-completeness of Kauffman's N–k model: a tunably rugged energy landscape, Technical Report 96-02-003, Santa Fe Institute, Santa Fe, NM, 1996.